Practical steps to mitigate DDoS attacks

Martin Čmelík

www.security-portal.cz

Security Session 2013, Brno, Czech Republic

What DoeS it mean?

- Exhausting resources like:
  - CPU
  - Memory/Buffers
  - I/O operations
  - Disk space
  - Network bandwidth
- Reach HW/SW or user defined limits (max. number of ...)
- Disruption of configuration or service crash

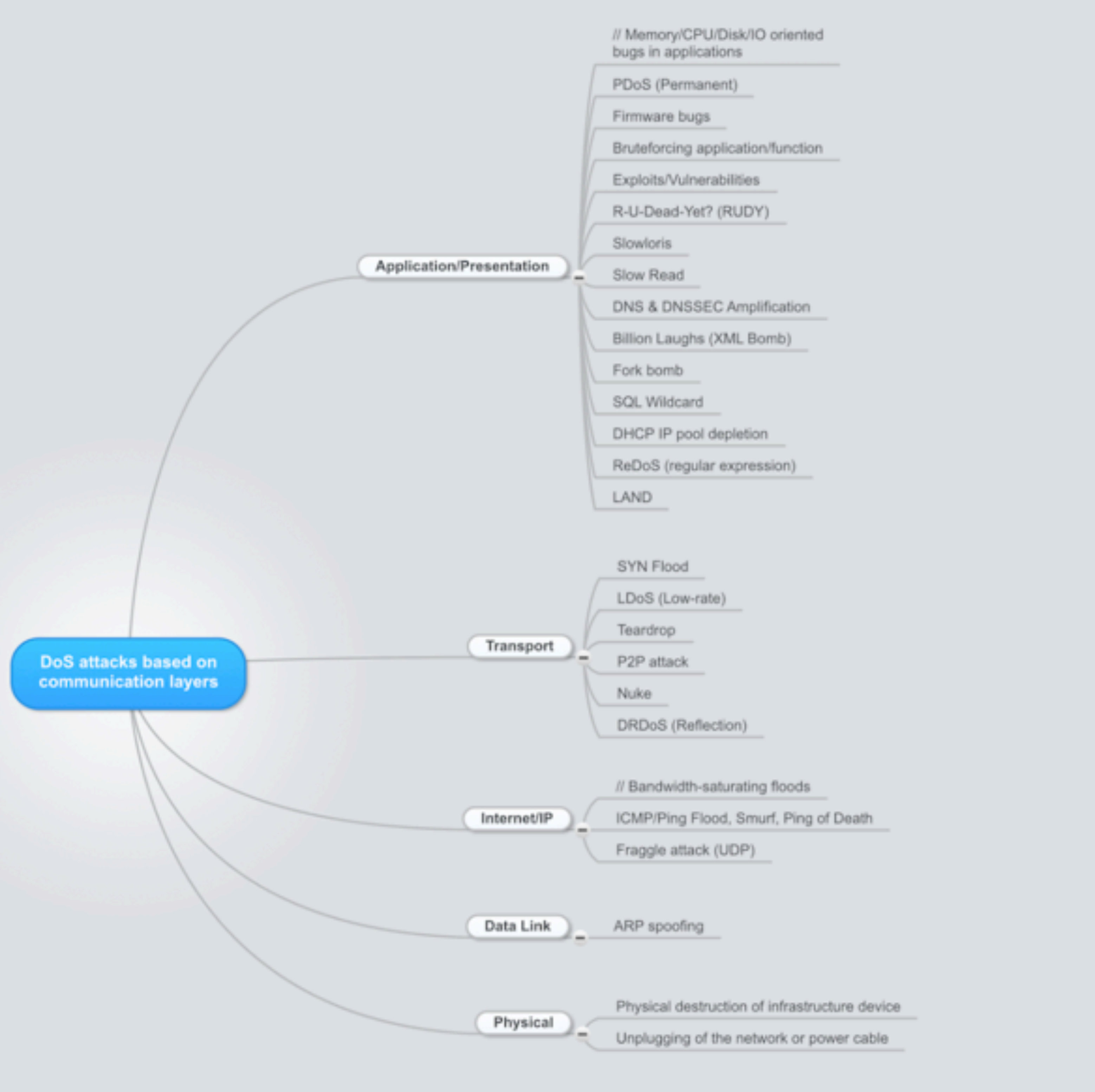
Attack motivation

- Financial gain
- Self-realization and social credit
- Revenge
- Political / Demonstration
- Cyber Terrorism
- Selling Anti-DDoS protection products
- To hide secondary attack

Factors improving DoS attacks

- Vulnerable systems
- Spoofing
- Existence of reflectors and amplifiers
- Data randomization (no signature)
- Bugs in applications
- Low bandwidth
- Lack of planning/testing/monitoring and risk management

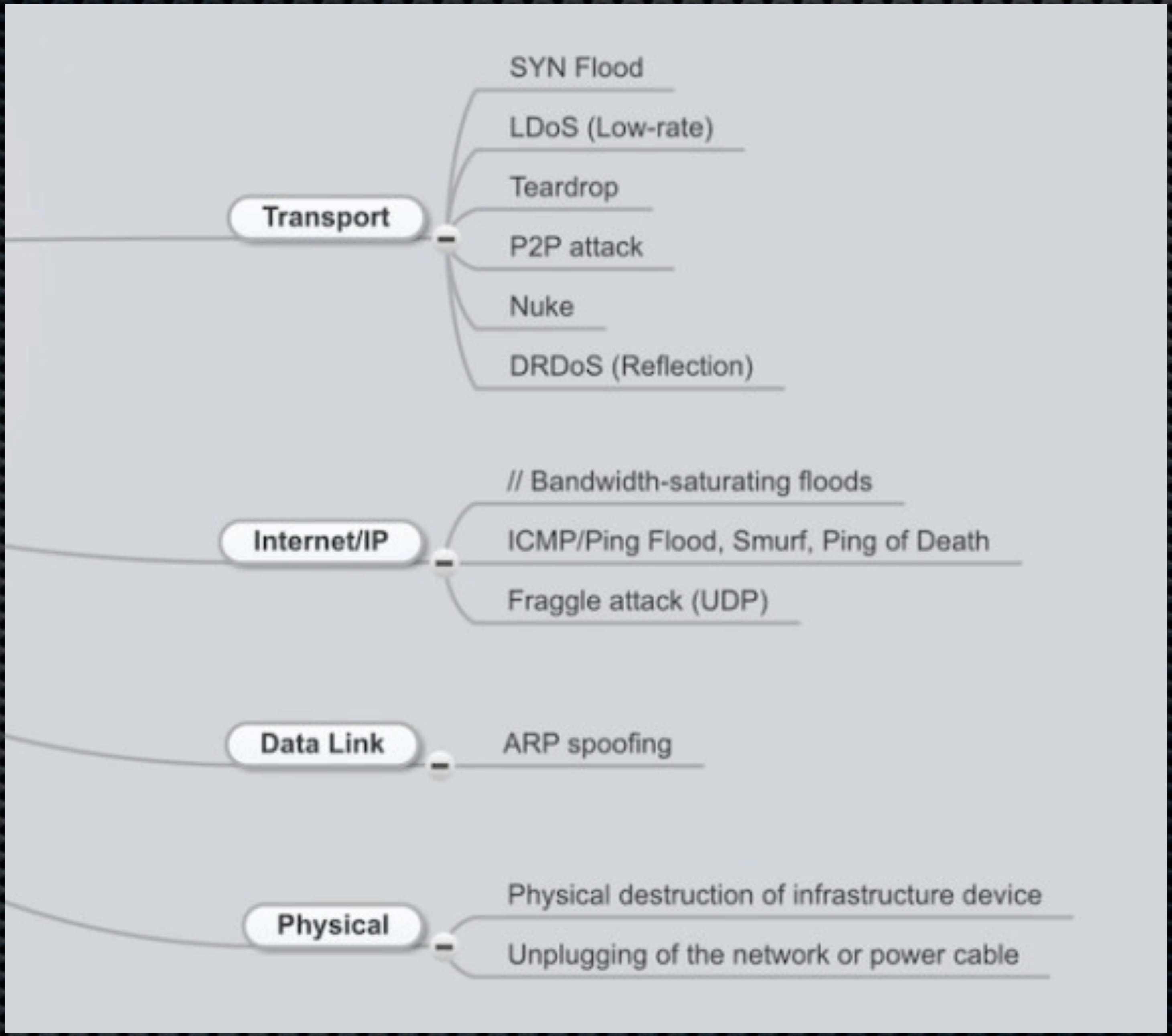
DoS attacks based on communication layers



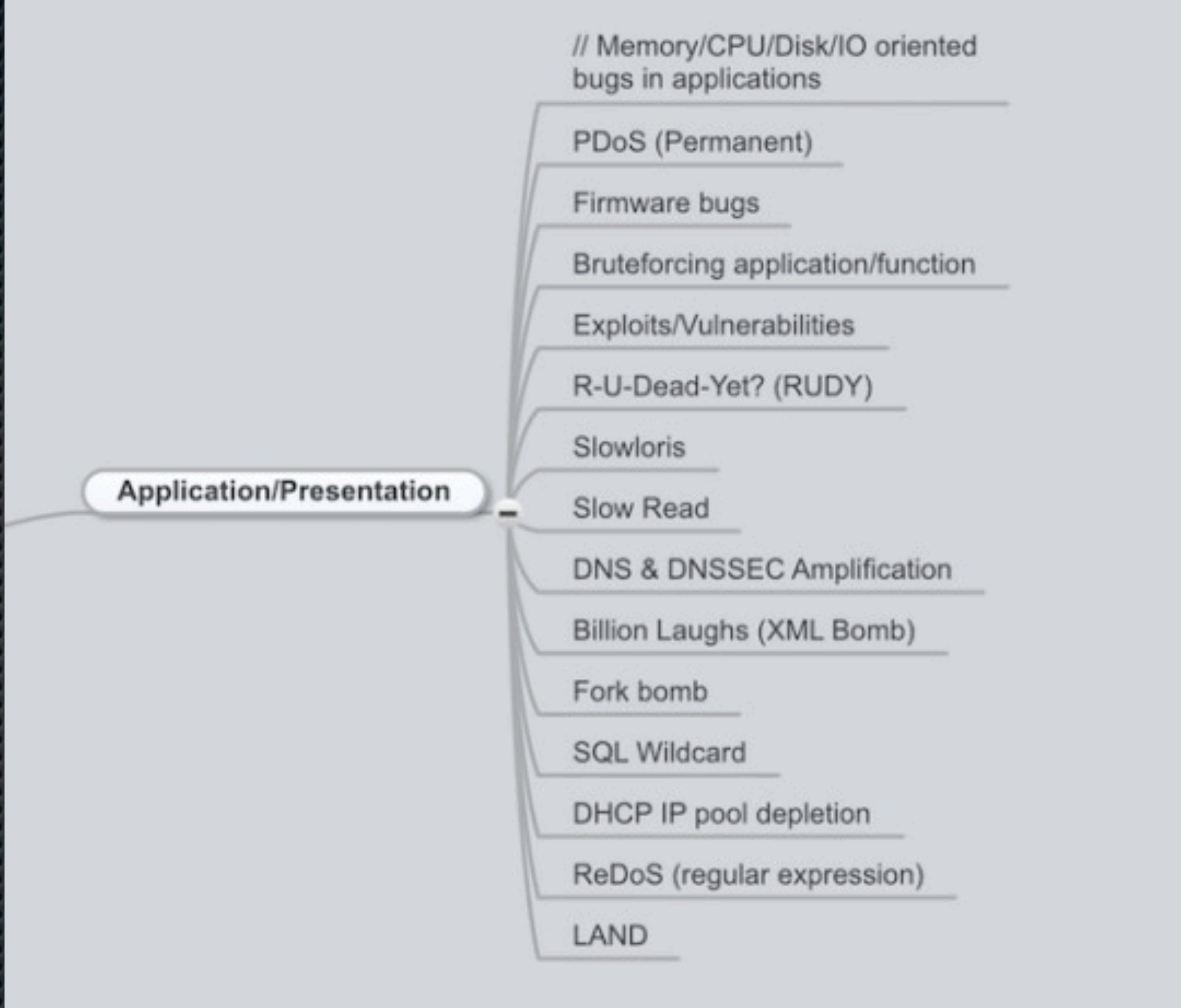
DoS attacks based on communication layers

- Application/Presentation
  - // Memory/CPU/Disk/IO oriented bugs in applications
  - PDoS (Permanent)
  - Firmware bugs
  - Bruteforcing application/function
  - Exploits/Vulnerabilities
  - R-U-Dead-Yet? (RUDY)
  - Slowloris
  - Slow Read
  - DNS & DNSSEC Amplification
  - Billion Laughs (XML Bomb)
  - Fork bomb
  - SQL Wildcard
  - DHCP IP pool depletion
  - ReDoS (regular expression)
  - LAND
- Transport
  - SYN Flood
  - LDoS (Low-rate)
  - Teardrop
  - P2P attack
  - Nuke
  - DRDoS (Reflection)
- Internet/IP
  - // Bandwidth-saturating floods
  - ICMP/Ping Flood, Smurf, Ping of Death
  - Fraggle attack (UDP)
- Data Link
  - ARP spoofing
- Physical
  - Physical destruction of infrastructure device
  - Unplugging of the network or power cable

DoS attacks - Transport, IP, data and physical layer

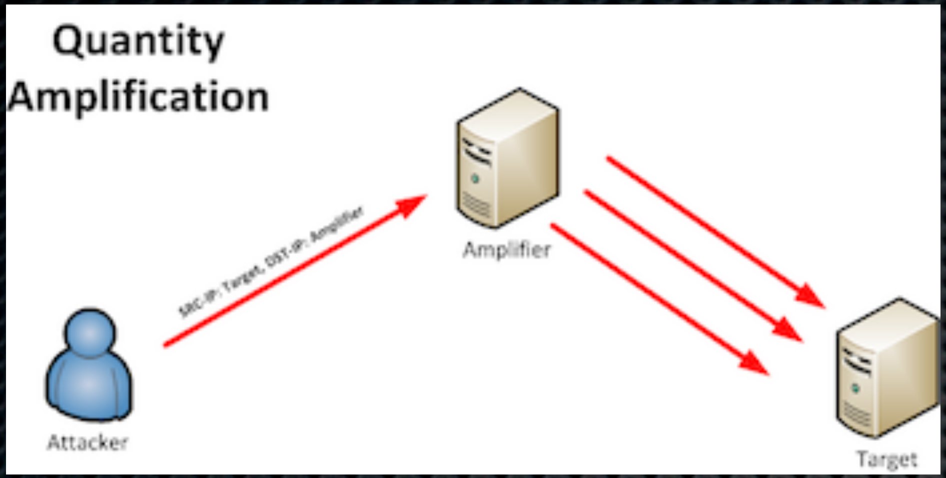

Transport
SYN Flood
LDoS (Low-rate)
Teardrop
P2P attack
Nuke
DRDoS (Reflection)
Internet/IP
// Bandwidth-saturating floods
ICMP/Ping Flood, Smurf, Ping of Death
Fraggle attack (UDP)
Data Link
ARP spoofing
Physical
Physical destruction of infrastructure device
Unplugging of the network or power cable

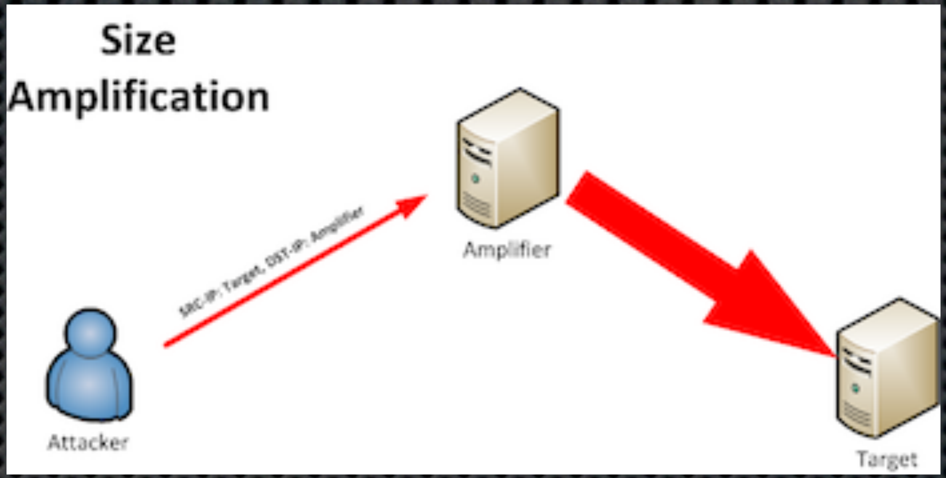
DoS attacks - Application and presentation layer

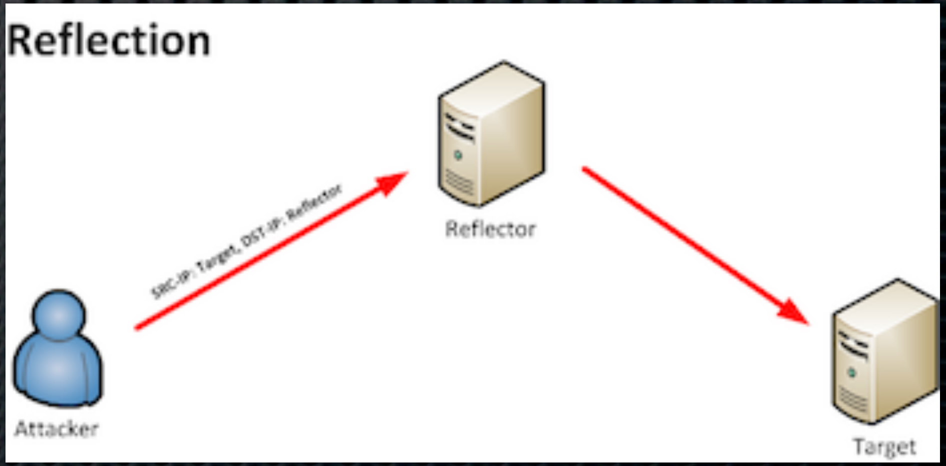


Application/Presentation

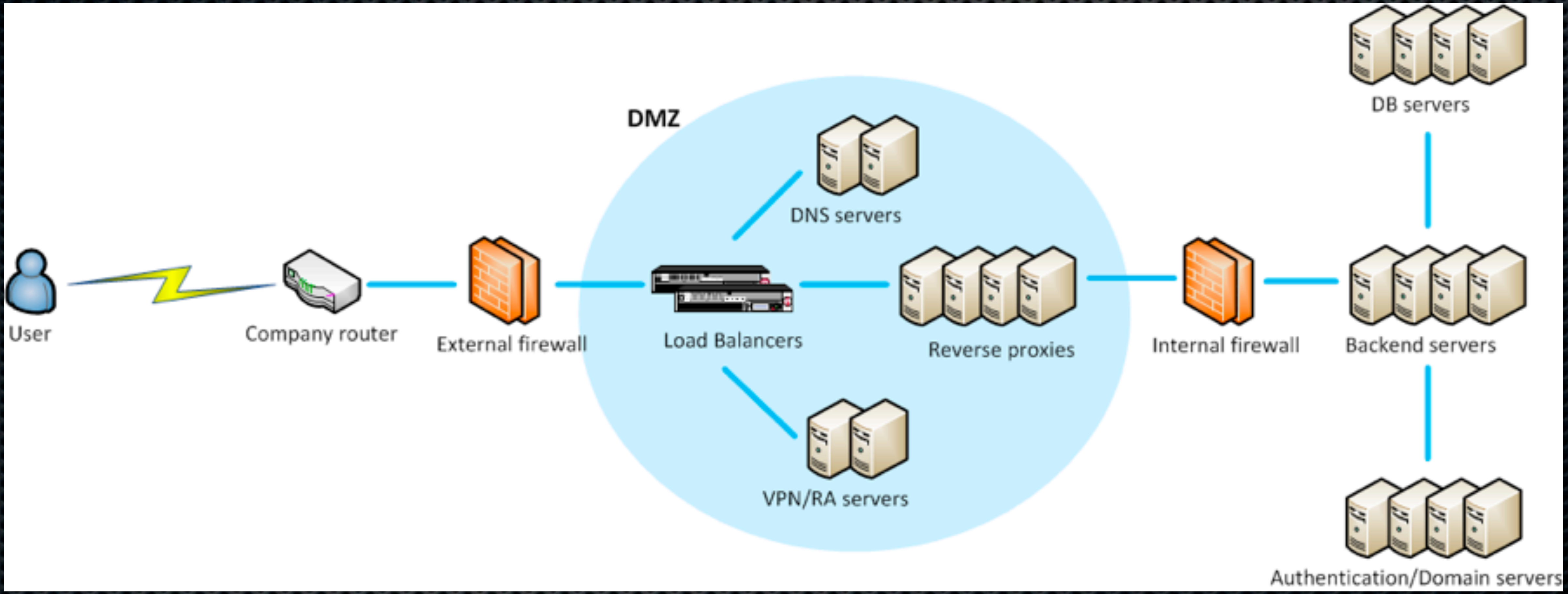
- // Memory/CPU/Disk/IO oriented bugs in applications
- PDoS (Permanent)
- Firmware bugs
- Bruteforcing application/function
- Exploits/Vulnerabilities
- R-U-Dead-Yet? (RUDY)
- Slowloris
- Slow Read
- DNS & DNSSEC Amplification
- Billion Laughs (XML Bomb)
- Fork bomb
- SQL Wildcard
- DHCP IP pool depletion
- ReDoS (regular expression)
- LAND


Quantity Amplification
SRC-IP: Target, DST-IP: Amplifier
Amplifier
Attacker
Target


Size Amplification
SRC-IP: Target, DST-IP: Amplifier
Amplifier
Attacker
Target


Reflection
SRC-IP: Target, DST-IP: Reflector
Reflector
Attacker
Target

Risk identification



Common recommendations

- Go through **all** devices on network, from L2 switches to backend servers and identify possible leaks, bottlenecks, attack vectors, applicable DoS attacks, vulnerabilities ... and mitigate or (rate)limit them
- Take our mind map and identify all applicable DoS attacks based on network layer where device operating
- In case of applications: identify time and resource consuming functions, make stress/fuzz testing and source code analysis
- Test your network and devices by simulating real DoS attack (LOIC/HOIC, hping, slowhttptest, thc-ssl-dos, pktgen, ...)

How to mitigate them?

JUNIPer NETWORKS DoS protection

- “In general, NetScreen offers DoS attack mitigation, hostile packet signature detection and blocking, and protection against port scans, address sweeps, and various flood attacks. In all current releases, NetScreen also offers a limited form of Malicious URL protection.”

// Source and Destination based session limits

```
# destination based limits can prevent amplification attack from internal DNS servers
set zone untrust screen limit-session destination-ip-based 4000
set zone untrust screen limit-session destination-ip-based
# source based can help against attempts to fill session table, or limit access
set zone dmz screen limit-session source-ip-based 1
set zone dmz screen limit-session source-ip-based
```

// Aggressive aging

Decrease timeouts for 3-way handshake (20s), TCP (30min), HTTP (5min), UDP (1min), ... when reached high-watermark threshold (for example 80%) and begins aggressively aging out the oldest sessions, until the number of sessions is under low-watermark. In this example timeouts will be decreased - 40 seconds from default.

```
set flow aging low-watermark 70
set flow aging high-watermark 80
set flow aging early-timeout 4
```

// CPU Protection with Blacklisting DoS attack traffic

Can create blacklists (up to 30) of IP addresses from which malicious traffic reach device and drop it without other processing inside device itself. This saves processing load on CPU during DoS

```
set cpu-protection blacklist id 1 1.1.1.0/24 2.2.2.0/24 protocol 17 src-
port 5 dst-port 7
timeout 0
```

// Prioritizing critical Traffic

Security device allows critical traffic (during high-utilization) such as management/routing/VPN/NSM traffic pass with priority and drops noncritical traffic. In this example activated when CPU is above 80%

```
set cpu-protection threshold 80
```

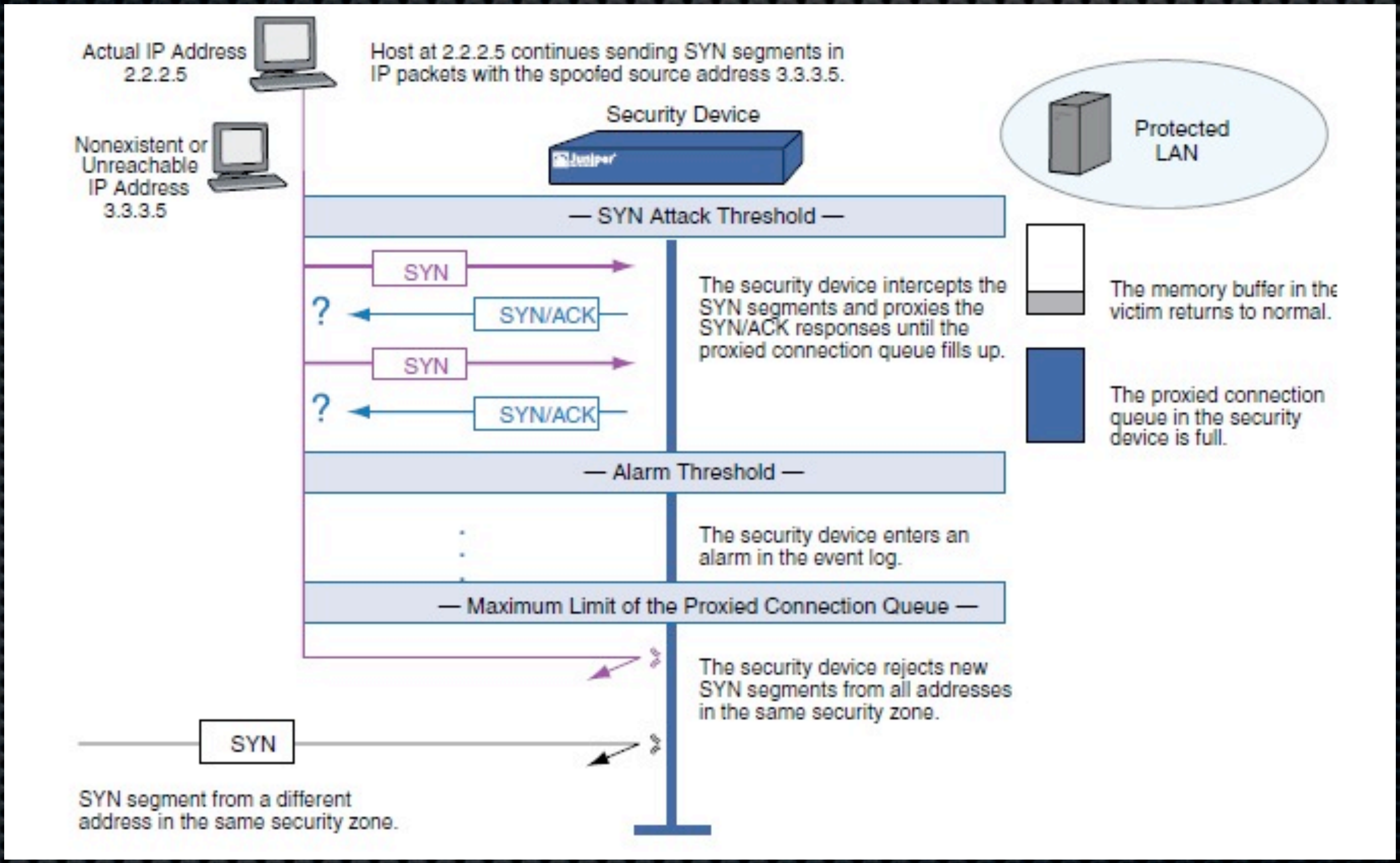
// SYN-ACK-ACK Proxy Flood

Prevent the SYN-ACK-ACK attack, which occurs when the attacker establishes multiple telnet sessions without allowing each session to terminate. This behavior consumes all open slots, generating a denial-of-service condition.

```
set zone untrust screen syn-ack-ack-proxy threshold 512
```

// SYN Flood protection

Limit the number of SYN pps based on source or destination IP address. When traffic exceeds threshold device will start proxying SYN packets. Same applies to Cisco embryonic connection protection and OpenBSD pf synproxy.


Actual IP Address 2.2.2.5
Host at 2.2.2.5 continues sending SYN segments in IP packets with the spoofed source address 3.3.3.5.
Security Device
Protected LAN
Nonexistent or Unreachable IP Address 3.3.3.5
— SYN Attack Threshold —
SYN
?
SYN/ACK
SYN
?
SYN/ACK
The security device intercepts the SYN segments and proxies the SYN/ACK responses until the proxied connection queue fills up.
The memory buffer in the victim returns to normal.
The proxied connection queue in the security device is full.
— Alarm Threshold —
The security device enters an alarm in the event log.
— Maximum Limit of the Proxied Connection Queue —
The security device rejects new SYN segments from all addresses in the same security zone.
SYN
SYN segment from a different address in the same security zone.

// SYN Flood protection - setup

```
# enable syn flood protection
set zone untrust screen syn-flood
set zone untrust screen syn-flood attack-threshold 1000
```

```
# alarm threshold here become effective when 3001 pps occur
set zone untrust screen syn-flood alarm-threshold 2000
set zone untrust screen syn-flood source-threshold 250
set zone untrust screen syn-flood destination-threshold 1000
```

```
# time until half-open connections are droped in queue (5s)

set zone untrust screen syn-flood timeout 5
```

```
# number of proxied connections before device starts rejecting new connection requests
set zone zone screen syn-flood queue-size 20000
```

// SYN cookies

Because SYN cookie is stateless, it does not set up a session or do policy and route lookups upon receipt of a SYN segment. This dramatically reduces CPU and memory usage and is main advantage instead of SYN proxying.

SYN cookie itself is computed ISN from first SYN packet: time, source IP and port, destination IP and port and MSS. When final ACK arrives (ISN+1) server knows to which cookie is that related.

In high-end devices, the PPU ASIC chip in the security device performs the SYN cookie mechanism instead of the security device CPU.

```
set zone untrust screen syn-flood
set zone untrust screen syn-flood attack-threshold 1000
set flow syn-proxy syn-cookie
```

// ICMP and UDP Flood protection

PING and UDP floods can have dramatic effect on firewall. In this example we will limit ICMP to 1000 pps and UDP to 10000 pps.

# enable ICMP flood protection

```
set zone untrust screen icmp-flood
set zone untrust screen icmp-flood threshold 1000
```

# block large ICMP

```
set zone untrust screen icmp-large
```

# enable UDP flood

```
set zone untrust screen udp-flood
set zone untrust screen udp-flood threshold 10000
```

// Land attack

Land attack occurs when an attacker sends spoofed SYN packets containing the IP address of the victim as both the destination and source IP address.

```
set zone untrust screen land
```

// Old types of DoS attacks

# Ping of Death (oversized ICMP packet)

```
set zone untrust screen ping-death
```

# Teardrop attack (reassembly of fragmented IP packets)

```
set zone untrust screen tear-drop
```

# WinNuke (NetBIOS packet with URG flag set -> BSOD)

```
set zone untrust screen winnuke
```

Check Point
SOFTWARE TECHNOLOGIES LTD.
We Secure the Internet.

DoS mitigation

- Applicable same improvements as for Linux
- SecureXL (traffic acceleration)
- CoreXL (balance rulebase processing across CPU cores)
- SIM Affinity (NICs IRQs balances across CPU cores)
- Global connection limit (by default only 25.000)
- Most of DoS attacks can be mitigated only by IPS module or DDoS Protector (upcoming)

SecureXL

- SecureXL is the security performance architecture which offloads many intensive security operations to optimized hardware or network processor. Offloaded security operations include TCP state negotiation, packet forwarding, Network Address Translation, VPN cryptography, anti-spoofing, routing and accounting.

# enable firewall acceleration

```
[Expert@internet-fw]# fwaccel on
```

# view statistics

```
[Expert@internet-fw]# fwaccel stats -s
Accelerated conns/Total conns : 68520/68706 (99%)
Accelerated pkts/Total pkts   : 8731331828/9006544387 (96%)
F2Fed pkts/Total pkts   : 275212333/9006544387 (3%)
PXL pkts/Total pkts   : 226/9006544387 (0%)
```

CoreXL

- This feature provides scalability of performance, according to the number of processor cores on a single machine. No change to network topology or management is required. CoreXL joins ClusterXL Load Sharing and SecureXL as part of the Check Point traffic acceleration technologies. The firewall kernel is replicated a number of times in a CoreXL gateway. Each instance or replicated copy of the firewall kernel runs on one processor core. These instances handle traffic concurrently, and each instance is a complete and independent inspection kernel.
- Enable via cpconfig option menu

CoreXL

When running CoreXL on four or more processing cores, the number of kernel instances in the CoreXL post-setup configuration is one less than the number of processing cores. The remaining processing core is responsible for processing incoming traffic from the network interfaces, securely accelerating authorized packets (if Performance Pack is running) and distributing non-accelerated packets among kernel instances.

#As you can see connections are balanced across CPU cores

```
[Expert@internet-fw]# fw ctl multik stat
ID | Active  | CPU | Connections | Peak
-------------------------------------------
 0 | Yes     | 7   |       11592 |    24596
 1 | Yes     | 6   |       13077 |    23970
 2 | Yes     | 5   |       10749 |    21975
 3 | Yes     | 4   |       10466 |    20683
 4 | Yes     | 3   |       12060 |    22448
 5 | Yes     | 2   |       12013 |    21772
```

SIM Affinity

This feature balances NICs IRQs across CPU cores. In ideal case each NIC will be assigned to unique CPU core. If it is not possible balance at least most utilized interfaces.

# Set automatic affinity

```
[Expert@internet-fw]# sim affinity -a
```

# Check that SIM affinity works

```
[Expert@internet-fw]# fw ctl affinity -l -v       # or # sim affinity -l
Interface Exp1-1 (irq 123): CPU 6
Interface Exp1-2 (irq 147): CPU 2
Interface Sync (irq 194): CPU 0
Interface Exp2-1 (irq 218): CPU 1
Interface Exp2-2 (irq 51): CPU 7
Interface Lan8 (irq 219): CPU 1
```

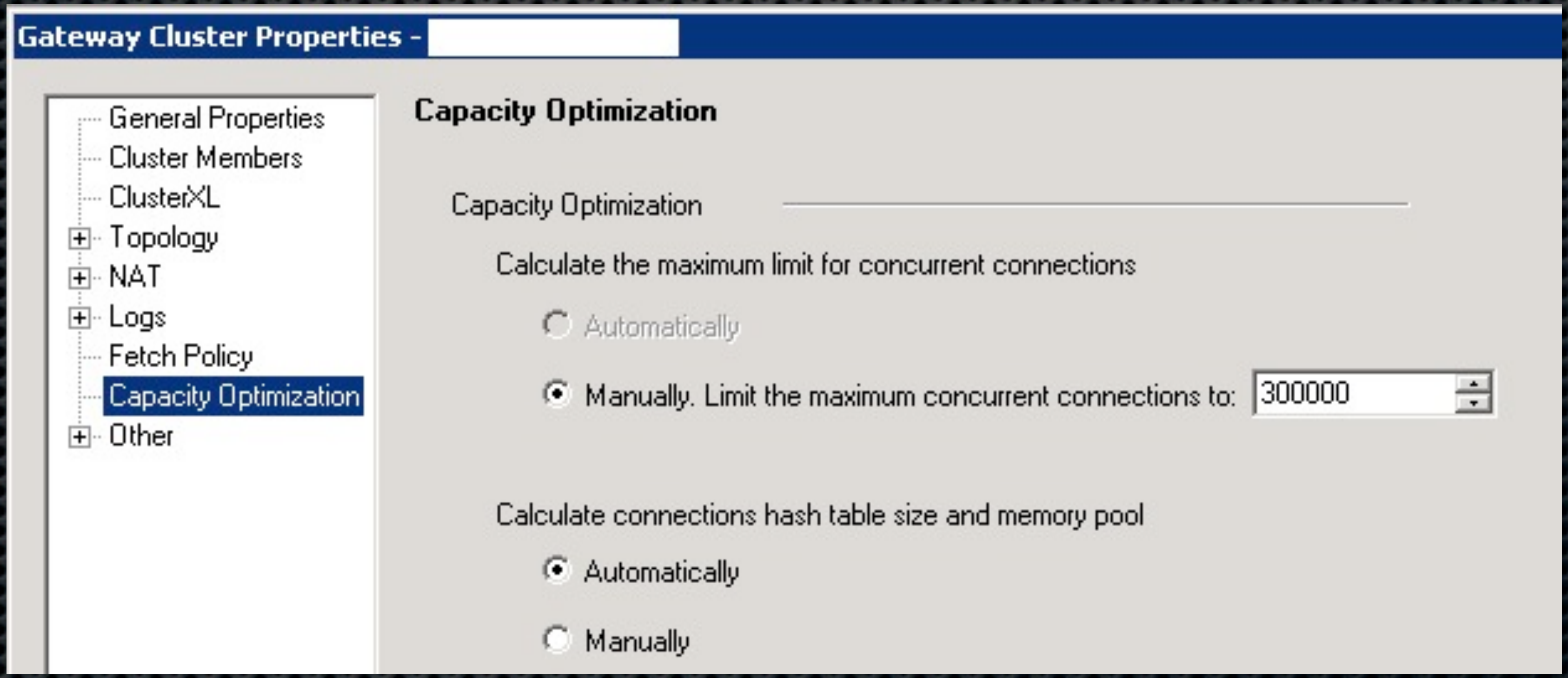
SIM Affinity

# Find most utilized interfaces

```
[Expert@internet-fw]# top
top - 13:08:48 up 606 days,  1:59,  1 user,  load average: 0.02,0.11, 0.09
Tasks: 122 total,   2 running, 119 sleeping,   0 stopped,   1 zombie
Cpu0  :  0.3%us,  0.3%sy,  0.0%ni, 91.7%id,  7.3%wa,  0.0%hi, 0.3%si,  0.0%st
Cpu1  :  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu2  :  0.0%us,  0.0%sy,  0.0%ni, 99.3%id,  0.0%wa,  0.0%hi,  0.7%si,  0.0%st
Cpu3  :  0.0%us,  0.3%sy,  0.0%ni, 98.3%id,  0.0%wa,  0.0%hi,  1.3%si,  0.0%st
Cpu4  :  0.0%us,  0.0%sy,  0.0%ni, 99.0%id,  0.0%wa,  0.3%hi,  0.7%si,  0.0%st
Cpu5  :  0.0%us,  0.0%sy,  0.0%ni, 95.3%id,  0.0%wa,  1.0%hi,  3.7%si,  0.0%st
Cpu6  :  4.0%us,  0.0%sy,  0.0%ni, 84.0%id,  0.0%wa,  1.3%hi, 10.7%si,  0.0%st
Cpu7  :  1.7%us,  0.0%sy,  0.0%ni, 81.6%id,  0.0%wa,  2.3%hi, 14.4%si,  0.0%st
Mem:   4150396k total,  2450612k used,  1699784k free,    82668k buffers
Swap:  8385920k total,       56k used,  8385864k free,  1781628k cached
```

Maximum concurrent connections

- Please be sure that you have increased maximum connections limit, which is by default 25.000 only and can be biggest bottleneck on your firewall.



Capacity Optimization

Capacity Optimization

Calculate the maximum limit for concurrent connections

○ Automatically

● Manually. Limit the maximum concurrent connections to: 300000

Calculate connections hash table size and memory pool

● Automatically

○ Manually

DoS mitigation

- Tune kernel parameters
- Increase NIC TX/RX buffers
- (D)DoS Deflate
- ...and you're safe :] (just kidding)

/etc/sysctl.conf tuning

```
# Decrease the time default value for tcp_fin_timeout connection
net.ipv4.tcp_fin_timeout = 15

# Decrease the time default value for tcp_keepalive_time connection
net.ipv4.tcp_keepalive_time = 1800

# Enable tcp_window_scaling
net.ipv4.tcp_window_scaling = 1

# Turn off the tcp_sack
net.ipv4.tcp_sack = 0

# Turn off the tcp_timestamps
net.ipv4.tcp_timestamps = 0

# This removes an odd behavior in the 2.6 kernels, whereby the
kernel stores the slow start threshold for a client between TCP
sessions.

net.ipv4.tcp_no_metrics_save = 1
```

/etc/sysctl.conf tuning

# Prevent SYN attack

```
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.tcp_syn_retries = 5
net.ipv4.tcp_synack_retries = 2
```

# Buffer size autotuning - buffer size (and tcp window size) is dynamically updated for each connection. This option is not present in kernels older then 2.4.27 or 2.6.7 - (update your kernel). In that case tuning options net.ipv4.tcp_wmem and net.ipv4.tcp_rmem isnt recommended

```
net.ipv4.tcp_moderate_rcvbuf = 1
```

# Increase the tcp-time-wait buckets pool size

```
net.ipv4.tcp_max_tw_buckets = 1440000
```

# Increase allowed local port range

```
net.ipv4.ip_local_port_range = 1024 64000
```

Increase TX/RX buffer size

If you have server under heavy (soft interrupts) load and interfaces aren't able to handle all incoming traffic, then RX-DRP occur, which in fact means: dropped, needs to be resend. If it is re-occurring issue client can recognize slowness of connection and intermittent disruption of service. In that case you have to check maximum allowed buffer size and increase it.

```
Linux ~ $ netstat -i
Kernel  Interface  table
Iface   MTU        Met    RX-OK       RX-ERR  RX-DRP   RX-OVR  TX-OK       TX-ERR
eth0    1500       0      1742981407  0       2607898  0       1176287425  0 0
lo      16436      0      126382      0       0        0       126382      0 0
```

# check current setup

```
Linux ~ $ ethtool -g eth0
Pre-set maximums:
RX: 4096
TX: 4096

Current hardware settings:
RX: 256
TX: 256
```

# set maximum values (put into rc scripts, settings will be lost after restart)

```
ethtool -G eth0 rx 4096 tx 4096
```

Simple connection limiting - (D)DoS Deflate

(D)DoS Deflate is a lightweight bash shell script designed to assist in the process of blocking a denial of service attack. It utilizes the command below to create a list of IP addresses connected to the server, along with their total number of connections. It is one of the simplest and easiest to install solutions at the software level.

```
netstat -ntu | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -n
```

IP addresses with over a pre-configured number of connections are automatically blocked in the server's firewall, which can be direct iptables or Advanced Policy Firewall (APF).

http://deflate.medialayer.com/

Apache DoS mitigation - mod_evasive

Module mainly utilized for rate-limiting. Can mitigate common HTTP flood attacks.

```
<IfModule mod_evasive20.c>

# size of hash table
DOSHashTableSize 4096
# requests for the _same_ page per interval and client
DOSPageCount 20
# requests for any object by same client
DOSSiteCount 300
# threshold in second intervals
DOSPageInterval 1
DOSSiteInterval 1
DOSBlockingPeriod 30
#DOSCloseSocket On
#DOSSystemCommand "/sbin/iptables -I INPUT -s %s -j DROP"
DOSWhitelist 127.0.0.1
DOSEmailNotify your@email.com
DOSLogDir /var/log/httpd/evasive.log

</IfModule>
```

Web Application DoS mitigation - mod_security

ModSecurity is open source web application firewall (WAF) operating as Apache/ Nginx/IIS module. WAFs are deployed to establish an external security layer that increases security, detects and prevents attacks (SQLi, XSS, LFI/RFI, ...) before they reach web applications. It provides protection from a range of attacks against web applications and allows for HTTP traffic monitoring and real-time analysis with little or no changes to existing infrastructure.

// OWASP Core rules

In order to provide generic web applications protection, the OWASP Core Rules use the following techniques:

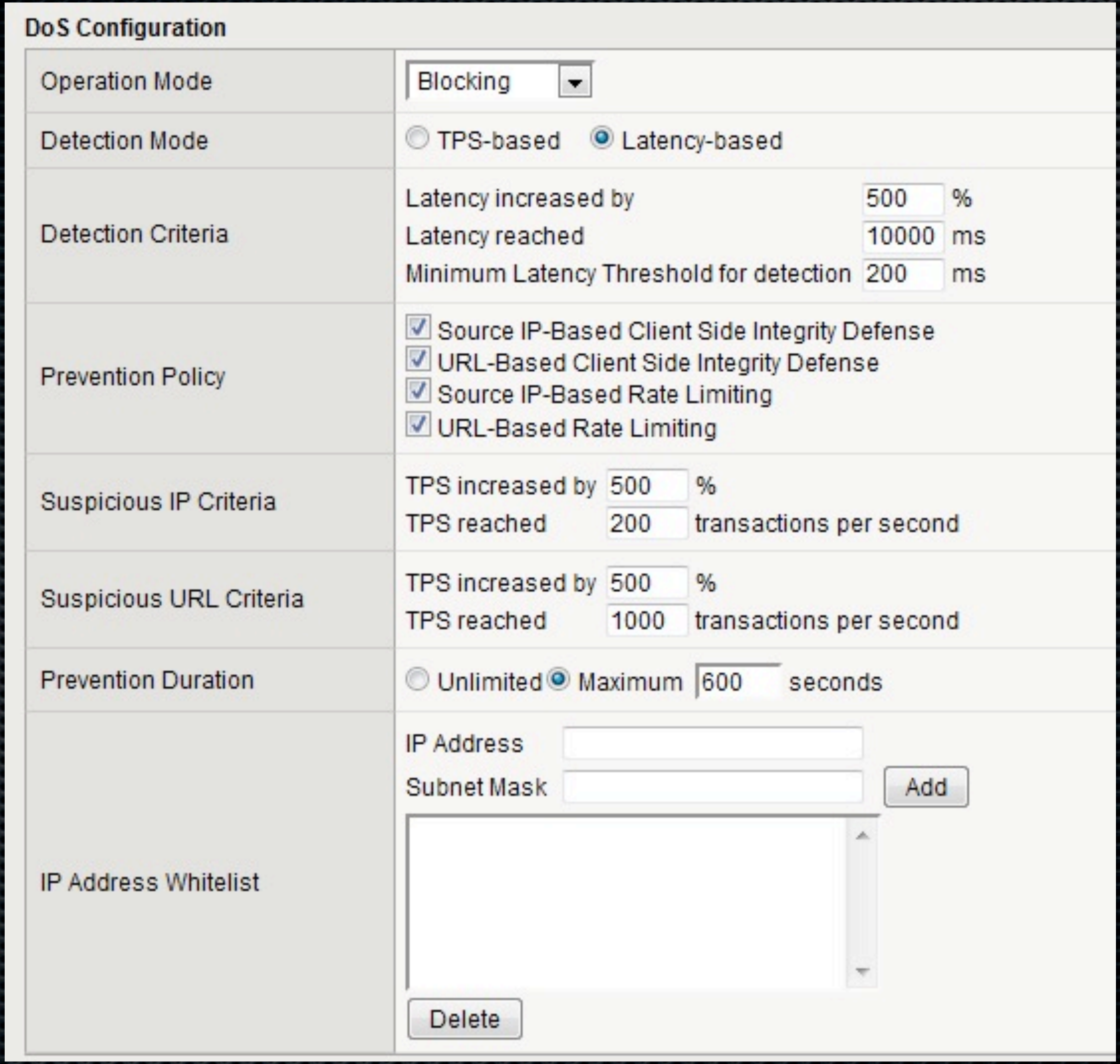
```
HTTP Protection (protocol violations)
Real-time Blacklist Lookups (3rd IP reputation)
Web-based Malware Detection (Google Safe Browsing API)
HTTP Denial of Service Protections (flooding, slow attacks)
Automation Detection (bots, crawlers, scanners)
Integration with AV Scanning for File Uploads
Tracking Sensitive Data (credit cards)
Trojan Protection
Identification of Application Defects
Error Detection and Hiding
```

https://www.owasp.org/index.pCategory:OWASP_ModSecurity_Core_Rule_Set_Project#Home

F5 BigIP Application Security Manager

- “F5 BIG-IP® Application Security Manager™ (ASM) is a flexible web application firewall that secures web applications in traditional, virtual, and private cloud environments. BIG-IP ASM provides unmatched web application and website protection, helps secure deployed applications against unknown vulnerabilities, and enables compliance for key regulatory mandates—all on a platform that consolidates application delivery with a data center firewall solution, and network and application access control.”

BigIP ASM DoS Attack Prevention



DoS Configuration

Operation Mode	Blocking ▾
Detection Mode	○ TPS-based ● Latency-based
Detection Criteria	Latency increased by 500 % Latency reached 10000 ms Minimum Latency Threshold for detection 200 ms
Prevention Policy	☑ Source IP-Based Client Side Integrity Defense ☑ URL-Based Client Side Integrity Defense ☑ Source IP-Based Rate Limiting ☑ URL-Based Rate Limiting
Suspicious IP Criteria	TPS increased by 500 % TPS reached 200 transactions per second
Suspicious URL Criteria	TPS increased by 500 % TPS reached 1000 transactions per second
Prevention Duration	○ Unlimited ● Maximum 600 seconds
IP Address Whitelist	IP Address Subnet Mask

BigIP ASM DoS Attack Prevention

- **Latency increased by**
  Specifies that the system considers traffic to be an attack if the latency has increased by this percentage.

  Latency reached
  Specifies that the system considers traffic to be an attack if the latency is equal to or greater than this value.

  Source IP-Based Client-Side Integrity Defense
  Checks whether a client is a legal browser or an illegal script by injecting JavaScript into responses when suspicious IP addresses are requested.

  URL-Based Client-Side Integrity Defense
  Checks whether a client is a legal browser or an illegal script by injecting JavaScript into responses when suspicious URLs are requested.

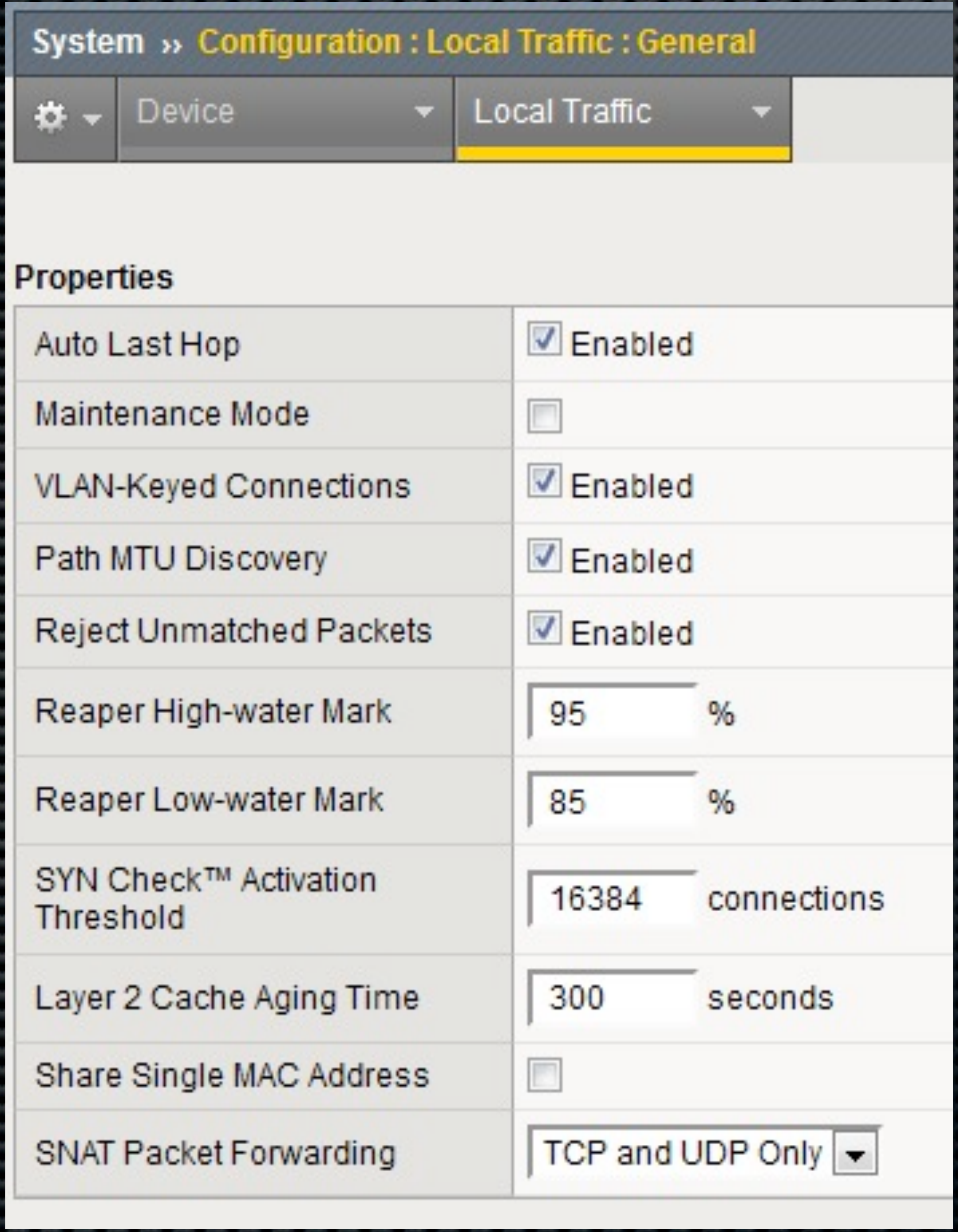
  Source IP-Based Rate Limiting
  Check to drop requests from suspicious IP addresses. Application Security Manager drops connections to limit the rate of requests to the average rate prior to the attack, or lower than the absolute threshold specified by the IP detection TPS reached setting.

  URL-Based Rate Limiting
  Check to indicate that when the system detects a URL under attack, Application Security Manager drops connections to limit the rate of requests to the URL to the average rate prior to the attack.

  Effective Challenge/Response authentication mechanism to differentiate attackers from normal users during DoS attack is usually JavaScript

BigIP LTM DoS attack prevention

System -> Configuration -> Local Traffic -> General



Properties

Auto Last Hop	☑ Enabled
Maintenance Mode	□
VLAN-Keyed Connections	☑ Enabled
Path MTU Discovery	☑ Enabled
Reject Unmatched Packets	☑ Enabled
Reaper High-water Mark	95 %
Reaper Low-water Mark	85 %
SYN Check™ Activation Threshold	16384 connections
Layer 2 Cache Aging Time	300 seconds
Share Single MAC Address	□
SNAT Packet Forwarding	TCP and UDP Only

BigIP LTM DoS attack prevention

- **Reaper High-water Mark**
  Specifies, in percent, the memory usage at which the system stops establishing new connections.
- **Reaper Low-water Mark**
  Specifies, in percent, the memory usage at which the system silently purges stale connections, without sending reset packets (RST) to the client.
- **SYN Check Activation Threshold**
  Specifies the number of new or untrusted TCP connections that can be established before the system activates the SYN Cookies.
- Please read following documentation for LTM DoS mitigation in detail (if needed)
  http://support.f5.com/kb/en-us/solutions/public/7000/300/sol7301.html

DNS Reflection and Amplification DoS prevention

- Don't setup DNS server as open resolver. A DNS resolver is open if it provides recursive name resolution for clients outside of its administrative domain.

DNS RRL is an experimental feature for domain name servers including CZ-NIC Knot DNS, NLNetLabs NSD, and ISC BIND9.

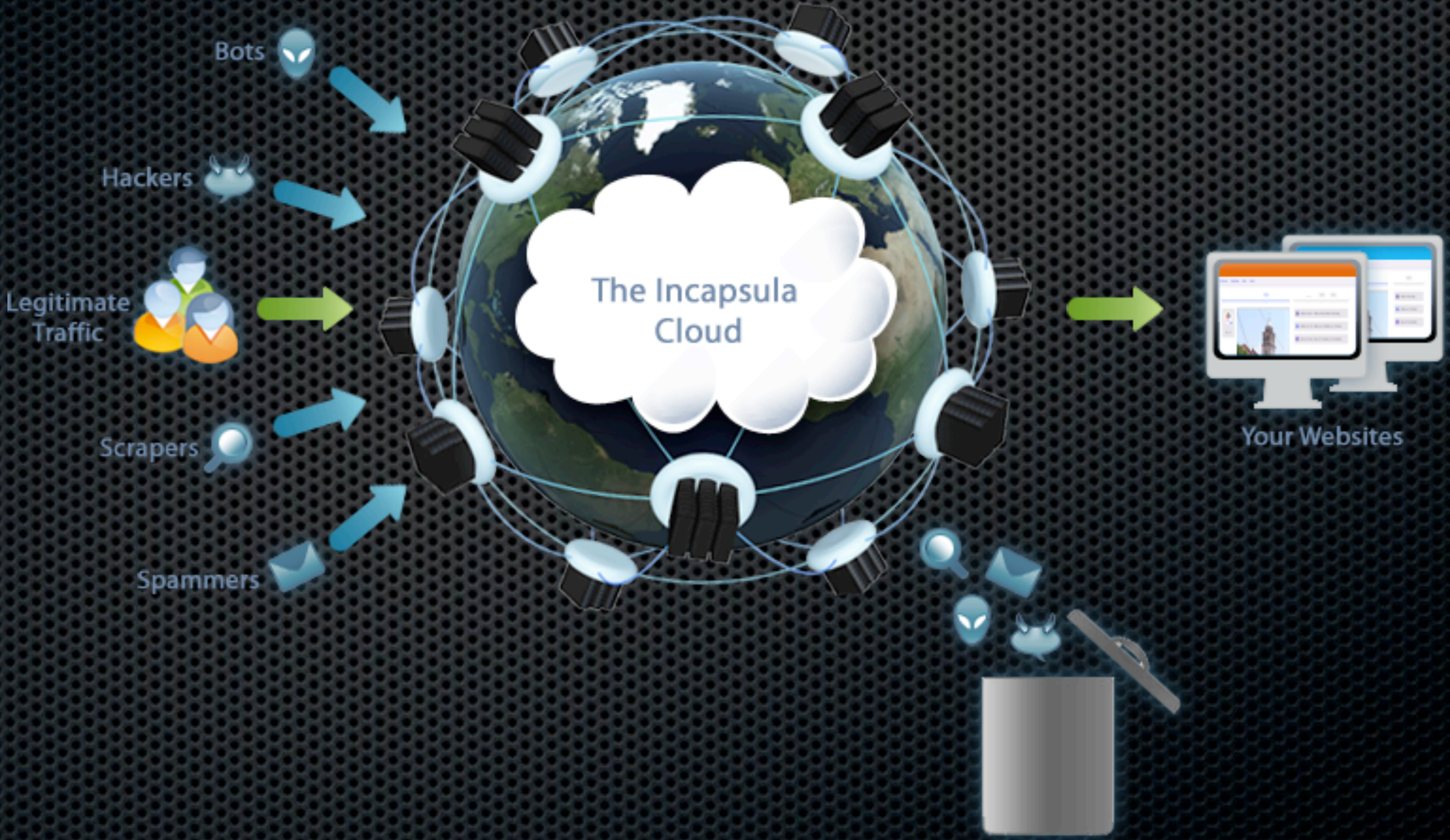
```
BIND example:
rate-limit {
    responses-per-second 5;
    window 5;
};

Knot example:
system {
    rate-limit 200;      # Each flow is allowed to 200 resp. per second
    rate-limit-slip 2;  # Every other response is slipped (default)
}
```

Why not utilizing TTL related limitation in combination with source IP based limit?

☁ based DoS protection

- Highly recommended to websites hosted on shared webhosting or VPS, but can be used on enterprise level as well. Relatively cheap and effective solution for most known DoS attacks.
- Take those risks on mind:
  - not all services can be protected, in most cases used for HTTP/S only
  - If you want HTTPS protection, SSL key pair will be sent to cloud provider and in fact they see unencrypted traffic (if not, your site can be subject of THC SSL DoS attack)
  - you must limit access to website only from cloud IP addresses, otherwise attacker can bypass cloud protection and access website directly

☁ based DoS protection



☁ based DoS protection

- Largest known DDoS attack peaking at 300Gb/s of traffic (Spamhaus). You're simply unable to block this amount of traffic on your Internet pipe. This is the biggest benefit of cloud based DoS protection. Cloud services can also protect you against application based attacks (XSS, SQLi, CSRF, Flooding, Slow attacks, protocol violations, ...) as described before in mode_security and thus looking like best choice. Don't forget that they can't limit all of them on your perimeter, that they have some limitations (often crucial) and can't protect your internal network or devices.
- Ask your ISP what he can do for you. What protections can offer (hopefully at least traffic blackholing - RTHB) and how to cooperate in case of DoS attack.

- **At the and I would like to recommend some providers**

  Dedicated DDoS mitigation equipment: `Arbor Networks, Cisco (CicoGuard), Toplayer, RioRey`

  Cloud protection for websites/small companies: `Incapsula, CloudFlare, Rivalhost, Imperva`

  Enterprise level cloud protection: `Akamai (DDoS Defender), Verisign, Prolexic, Gigenet, Staminus`

  Security of your company is in your hands. DoS/DDoS is only one of many (often more dangerous) attacks.

Thank you!

Martin Čmelík

www.linkedin.com/in/martincmelik

www.security-portal.cz | www.securix.org | www.security-session.cz